IN THE CLAIMS:

Please write the claims to read as follows:

Please cancel claims 26-38 without prejudice.

1. (Original) A method for defragmenting data blocks on disks of a computer configured to implement a file system that logically organizes the blocks as a file on the disks, the file further including indirect blocks having pointers that reference the data blocks, the method comprising the steps of:

determining a current layout of a range of pointers contained in each indirect block of the file;

calculating a number of operations needed to retrieve the data blocks referenced by the pointers from the disks to a memory of the computer;

estimating a potential new layout based on an average fullness of the file system; and

relocating the data blocks on the disks if the potential new layout improves the current layout.

2. (Original) The method of Claim 1 wherein the step of relocating comprises the step of relocating the data blocks if there is sufficient free space on the disks.

3. (Original) The method of Claim 2 wherein the step of relocating comprises the steps of loading the data blocks into the memory of the computer and dirtying the data blocks.

4. (Original) The method of Claim 3 wherein the step of relocating further comprises the steps of:

searching a predetermined distance of a first disk for free blocks; and

filling those free blocks with the dirtied data blocks.

5. (Original) The method of Claim 4 wherein the step of relocating further comprises the steps of:

jumping to a second disk;

searching the predetermined distance of the second disk for additional free blocks; and

filling those additional free blocks with the dirtied data blocks.

6. (Original) The method of Claim 5 wherein the step of relocating further comprises the step of repeating the steps of jumping, searching and filling until all data blocks of the file have been relocated.

7. (Original) The method of Claim 5 wherein the predetermined distance is 32 data blocks.

8. (Original) A system adapted to defragment data blocks on disks of a computer configured to implement a file system that logically organizes the blocks as a file on the disks, the file further including indirect blocks having pointers that reference the data blocks, the system comprising:

a processor coupled to the disks;

a memory coupled to the processor and having locations addressable by the processor; and

a storage operating system resident in the memory locations and executed by the processor to invoke storage operations in support of the file system, the storage operating system including a scanner adapted to (i) determine a current layout of a range of pointers contained in each indirect block of the file, (ii) calculate a number of operations needed to retrieve the data blocks referencing the pointers from the disks to the memory and (iii) estimate a potential new layout based on an average fullness of the file system, the storage operating system further including a write allocator adapted to relocate the data blocks on the disks if the potential new layout improves fragmentation of the current layout.

9. (Original) The system of Claim 8 wherein the file system is a write anywhere file system.

10. (Original) The system of Claim 9 wherein the scanner comprises a defragmentation process.

11. (Original) The system of Claim 10 wherein the range of pointers is a number of data blocks referenced by the pointers of an indirect block.

12. (Original) The system of Claim 11 wherein the number of data blocks referenced by the pointers of an indirect block is 1024.

13. (Original) The system of Claim 8 wherein the memory is a buffer cache.

14. (Previously Presented) A method for defragmenting data blocks on disks of a computer configured to implement a file system that logically organizes the blocks as a file on the disks, the file further including indirect blocks having pointers that reference the data blocks, the method comprising the steps of:

locating a beginning of the file;

selecting a range of data blocks to defragment, where the range of data blocks is a number of blocks referenced by an indirect block;

attempting defragmentation of the range of data blocks by determining whether a new layout is better then an existing layout; and

repeating the steps of selecting and attempting until defragmentation of all ranges of data blocks within the file has been attempted.

15. (Canceled)

15

~~16~~. (Previously Presented) The method of Claim 14 wherein the number of blocks is 1024 blocks.

16

~~17~~. (Previously Presented) A method for defragmenting data blocks on disks of a computer configured to implement a file system that logically organizes the blocks as a file on the disks, the file further including indirect blocks having pointers that reference the data blocks, the method comprising the steps of:

locating a beginning of the file;

selecting a range of data blocks to defragment, where the range of data blocks is a number of blocks referenced by an indirect block;

attempting defragmentation of the range of data blocks comprising,

reading pointers for the range of data blocks,

measuring an existing layout of the range of data blocks,

estimating a new layout of the range of data blocks,

determining whether the new layout is better than the existing layout,

if the new layout is better, determining whether there is sufficient free space in the file system to relocate the data blocks,

if there is sufficient free space, reading the data blocks into a buffer cache and dirtying the data blocks, and

rewriting the dirtied data blocks to new locations on the disks; and

repeating the steps of selecting and attempting until defragmentation of all ranges of data blocks within the file has been attempted.

17 16

~~18~~. (Original) The method of Claim ~~17~~ wherein the step of measuring comprises the step of calculating a number of write_alloc_chunks needed to cover the data blocks in the range.

18 17

~~19~~. (Original) The method of Claim ~~18~~ wherein the number of write_alloc_chunks ranges from a number of data blocks in the range and a number of data blocks in the range divided by a write_alloc_chunk.

19 17

~~20~~. (Original) The method of Claim ~~18~~ wherein the step of estimating comprises the step of estimating the new layout using an average fullness of the file system.

20 17

21. (Original) The method of Claim 18 wherein the step of determining whether the new layout is better than the existing layout comprises the step of comparing existing and estimated numbers of write_alloc_chunks.

21

22. (Original) Apparatus for defragmenting data blocks on disks of a computer configured to implement a file system that logically organizes the blocks as a file on the disks, the file further including indirect blocks having pointers that reference the data blocks, the apparatus comprising:

means for determining a current layout of a range of pointers contained in each indirect block of the file;

means for calculating a number of operations needed to retrieve the data blocks referenced by the pointers from the disks to a memory of the computer;

means for estimating a potential new layout based on an average fullness of the file system; and

means for relocating the data blocks on the disks if the potential new layout improves the current layout.

22 21

23. (Original) The apparatus of Claim 22 wherein the means for relocating comprises:

means for searching a predetermined distance of the disks for free blocks; and

means for filling the free blocks with the data blocks.

23

24. (Original) A computer readable medium containing executable program instructions for defragmenting data blocks on disks of a computer configured to implement a file system that logically organizes the blocks as a file on the disks, the file further including

indirect blocks having pointers that reference the data blocks, the executable program instructions comprising program instructions for:

determining a current layout of a range of pointers contained in each indirect block of the file;

calculating a number of operations needed to retrieve the data blocks referenced by the pointers from the disks to a memory of the computer;

estimating a potential new layout based on an average fullness of the file system; and

relocating the data blocks on the disks if the potential new layout improves the current layout.

24 23
25. (Original) The computer readable medium of Claim 24 wherein the program instruction for relocating comprises program instructions for:

searching a predetermined distance of the disks for free blocks; and

filling the free blocks with the data blocks.

26-38. (Canceled)